

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): MATSUI, ET AL.
Serial No.: Not assigned
Filed: June 19, 2001
Title: PARALLEL PROCESSES RUN SCHEDULING METHOD AND
DEVICE AND COMPUTER READABLE MEDIUM HAVING A
PARALLEL PROCESSES RUN SCHEDULING PROGRAM
RECORDED THEREON
Group: Not assigned



LETTER CLAIMING RIGHT OF PRIORITY

Honorable Commissioner of
Patents and Trademarks
Washington, D.C. 20231

June 19, 2001

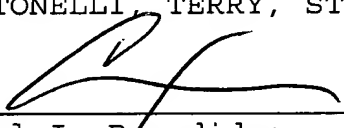
Sir:

Under the provisions of 35 USC 119 and 37 CFR 1.55, the applicant(s) hereby claim(s) the right of priority based on Japanese Patent Application No.(s) 2000-182554, filed June 19, 2000.

A certified copy of said Japanese Application is attached.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP



Carl I. Brundidge
Registration No. 29,621

CIB/amr
Attachment
(703) 312-6600

日 本 国 特 許 庁
JAPAN PATENT OFFICE

JC997 U.S. PTO
09/883173



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日
Date of Application:

2000年 6月19日

出 願 番 号
Application Number:

特願2000-182554

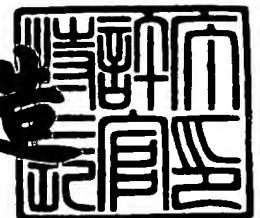
出 願 人
Applicant(s):

株式会社日立製作所

2001年 5月30日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3046397

【書類名】 特許願

【整理番号】 K000227I

【提出日】 平成12年 6月19日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 15/16

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社
日立製作所 ソフトウェア開発本部内

【氏名】 松井 謙治

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社
日立製作所 ソフトウェア開発本部内

【氏名】 戸部 和政

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社
日立製作所 ソフトウェア開発本部内

【氏名】 永田 祐彦

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社
日立製作所 ソフトウェア開発本部内

【氏名】 熊▲崎▼ 裕之

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100099298

【弁理士】

【氏名又は名称】 伊藤 修

【連絡先】 0 3 - 3 2 5 1 - 3 8 2 4

【選任した代理人】

【識別番号】 100099302

【弁理士】

【氏名又は名称】 笹岡 茂

【手数料の表示】

【予納台帳番号】 018647

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 プロセスのスケジューリング方法および装置およびプロセスのスケジューリングプログラムを記録したコンピュータ読み取り可能な記録媒体

【特許請求の範囲】

【請求項 1】 複数のプロセッサ上で動作する並列度が等しい複数の並列プログラムを時分割により多重に実行する情報処理システムにおけるプロセスのスケジューリング方法であって、

各プロセッサが、全てのプロセッサで同期して積算されるカウンタ値と並列プログラムの多重度から、それぞれ実行する各並列プロセスの実行順序を決定することにより、同一の並列プログラムの全ての並列プロセスを同時に実行することを特徴とするプロセスのスケジューリング方法。

【請求項 2】 複数のプロセッサ上で動作する並列度が異なる複数の並列プログラムを時分割により多重に実行する情報処理システムにおけるプロセスのスケジューリング方法であって、

各プロセッサが、全てのプロセッサで同期して積算されるカウンタ値と並列プログラムの多重度から、それぞれ実行する各並列プロセスの実行順序を決定し、

並列プログラムの並列度が時分割により多重に実行する際に使用するプロセッサ数より少ない場合には、該並列プログラムの並列プロセスを割り当てられていないプロセッサは、擬似プロセスを該並列プログラムの実行時に同時に実行することにより、同一の並列プログラムの並列プロセスを同時に実行することを特徴とするプロセスのスケジューリング方法。

【請求項 3】 複数のプロセッサ上で動作する並列度が等しい複数の並列プログラムを時分割により多重に実行する情報処理システムにおけるプロセスのスケジューリング装置であって、

前記各プロセッサは、全てのプロセッサで同期して積算される積算手段と、

実行する並列プログラムの数に対応するプロセス数を格納するプロセス数格納手段と、

実行するプロセスのプロセスキューを生成する手段と、

該生成したプロセスキューを格納するプロセスキュー格納手段と、

該プロセスキューの内の実行するプロセスを前記積算手段の積算値と前記プロセス数格納手段のプロセス数とにより決定する実行プロセス決定手段を有することを特徴とするプロセスのスケジューリング装置。

【請求項4】 複数のプロセッサ上で動作する並列度が異なる複数の並列プログラムを時分割により多重に実行する情報処理システムにおけるプロセスのスケジューリング装置であって、

前記各プロセッサは、全てのプロセッサで同期して積算される積算手段と、

実行する並列プログラムの数に対応するプロセス数を格納するプロセス数格納手段と、

実行するプロセス数が該プロセス数格納手段内のプロセス数より小さい場合は該両プロセス数の差の数だけの疑似プロセスを含む実行するプロセスのプロセスキューを生成し、該両プロセス数が等しい場合には疑似プロセスを含まない実行するプロセスのプロセスキューを生成する手段と、

該生成したプロセスキューを格納するプロセスキュー格納手段と、

該プロセスキューの内の実行するプロセスを前記積算手段の積算値と前記プロセス数格納手段のプロセス数とにより決定する実行プロセス決定手段を有することを特徴とするプロセスのスケジューリング装置。

【請求項5】 指定された実行する並列プログラムの数に対応するプロセス数をカウントし記憶する手順と、

指定された実行するプロセスのプロセスキューを生成し記憶する手順と、

該プロセスキュー内のプロセスの内の実行するプロセスをプロセッサの積算手段の積算値と前記並列プログラムの数に対応するプロセス数とにより決定する手順を有するプロセスのスケジューリングプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項6】 指定された実行する並列プログラムの数に対応するプロセス数をカウントし記憶する手順と、

指定された実行するプロセス数が前記カウントされたプロセス数より小さい場合は該両プロセス数の差の数だけの疑似プロセスを含むプロセスキューを生成し、該両プロセス数が等しい場合には疑似プロセスを含まないプロセスのプロセス

キューを生成し記憶する手順と、

該プロセスキュー内のプロセスの内の実行するプロセスをプロセッサの積算手段の積算値と前記並列プログラムの数に対応するプロセス数とにより決定する手順を有するプロセスのスケジューリングプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、複数のプロセッサを有し、それらを結合網により接続した並列コンピュータシステムにおいて、複数のプロセッサ上で動作する並列プログラムの並列プロセスを、時分割によって多重に実行する情報処理システムにおけるプロセスのスケジューリングに関する。

【0002】

【従来の技術】

並列コンピュータシステムでは、複数のプロセッサと、それらを結合する結合網を有している。

複数のプロセッサを用いて実行される並列プログラムは、いくつかの並列プロセスに分割され、各々、別のプロセッサ上で並列に動作する。

複数のプロセッサ上で並列に動作する該プロセスは、結合網を通じて、互いにデータをやりとりしながら、全体として一つの結果を導き出す。

このように実行される並列プロセスでは、あるチェックポイントごとに同期して実行が行われる。

このとき、相手プロセスが、そのチェックポイントに達していない場合、相手プロセスとのデータのやりとりが行えず、いわゆる同期待ちが生じる。

【0003】

また、福田晃著「並列オペレーティングシステム」（コロナ社、1997年発行）（ISBN4-339-0258-9）の第55頁から第56頁に示してあるように、複数の並列プログラムを同時に実行する場合、コンピュータシステム全体を、時分割によって利用する方法がある。

並列コンピュータシステムで、複数の並列プログラムが時分割によって多重に実行される場合、各プロセッサ上で、プロセスのスケジューリングを行い、プロセスの切換えを行う必要がある。

このときに、複数のプロセッサ間の相互で協調することなく並列プロセスをスケジューリングすると、各プロセッサで実行される並列プロセスの実行時間差のために、並列プロセスの同期待ち時間が大きくなるという課題がある。

この同期待ち時間の増大を避けるために、並列プログラムの並列プロセスのスケジューリングでは、同一の並列プログラムの並列プロセスを同時に実行するように、プロセッサ間の相互で協調したプロセスのスケジューリングを行う必要がある。

【 0 0 0 4 】

このように、プロセッサ間の相互で協調したスケジューリングを行う方法として、(1) 福田晃著「並列オペレーティングシステム」(コロナ社, 1997年発行) (ISBN 4-339-0258-9) の第56頁から第62頁では、並列プログラムの並列プロセスをタイムスライスの間同時実行し、その後、グローバルなスケジューラが呼ばれ、次に実行するプロセスを選び、各プロセッサに通知する、プロセスのスケジューリング方法を示している。

また、(2) 公開特許公報特開平 10-74150号では、複数のプロセッサ上で並列プロセスを実行するため、一定期間のプロセスの開始、停止時刻をあらかじめ決定したプログラム割当て表を作成し、各プロセッサに通知しておき、各プロセッサでは、それにしたがってプロセスを実行する、プロセスのスケジューリング方法を示している。

【 0 0 0 5 】

【発明が解決しようとする課題】

しかしながら、上記のようなスケジューリング方法では、(1) ではプロセスの切換え毎に、(2) ではプログラム割当て表の作成毎にプロセッサに通知する処理が必要となる。

そのため、各プロセッサで行う処理の他に、プロセッサ間での通知の処理が必要になり、従来のプロセッサの相互で協調したスケジューリングでは、該処理の

オーバヘッド時間が問題となる。

本発明の目的は、プロセッサの相互で協調したスケジューリングに伴うオーバヘッドを削減した、プロセッサ間の明示的な通信を必要としない、プロセスのスケジューリング方法を提供することにある。

【 0 0 0 6】

【課題を解決するための手段】

上記目的を達成するため、本発明は、

複数のプロセッサ上で動作する並列度が等しい複数の並列プログラムを時分割により多重に実行する情報処理システムにおけるプロセスのスケジューリング方法であり、

各プロセッサが、全てのプロセッサで同期して積算されるカウンタ値と並列プログラムの多重度から、それぞれ実行する各並列プロセスの実行順序を決定することにより、同一の並列プログラムの全ての並列プロセスを同時に実行するようにしている。

また、複数のプロセッサ上で動作する並列度が異なる複数の並列プログラムを時分割により多重に実行する情報処理システムにおけるプロセスのスケジューリング方法であり、

各プロセッサが、全てのプロセッサで同期して積算されるカウンタ値と並列プログラムの多重度から、それぞれ実行する各並列プロセスの実行順序を決定し、

並列プログラムの並列度が時分割により多重に実行する際に使用するプロセッサ数より少ない場合には、該並列プログラムの並列プロセスを割り当てられていないプロセッサは、擬似プロセスを該並列プログラムの実行時に同時に実行することにより、同一の並列プログラムのすべての並列プロセスを同時に実行するようにしている。

また、複数のプロセッサ上で動作する並列度が等しい複数の並列プログラムを時分割により多重に実行する情報処理システムにおけるプロセスのスケジューリング装置であり、

前記各プロセッサは、全てのプロセッサで同期して積算される積算手段と、

実行する並列プログラムの数に対応するプロセス数を格納するプロセス数格納

手段と、

実行するプロセスのプロセスキューを生成する手段と、

該生成したプロセスキューを格納するプロセスキュー格納手段と、

該プロセスキューの内の実行するプロセスを前記積算手段の積算値と前記プロセス数格納手段のプロセス数とにより決定する実行プロセス決定手段を有するようにしている。

また、複数のプロセッサ上で動作する並列度が異なる複数の並列プログラムを時分割により多重に実行する情報処理システムにおけるプロセスのスケジューリング装置であり、

前記各プロセッサは、全てのプロセッサで同期して積算される積算手段と、

実行する並列プログラムの数に対応するプロセス数を格納するプロセス数格納手段と、

実行するプロセス数が該プロセス数格納手段内のプロセス数より小さい場合は該両プロセス数の差の数だけの疑似プロセスを含む実行するプロセスのプロセスキューを生成し、該両プロセス数が等しい場合には疑似プロセスを含まない実行するプロセスのプロセスキューを生成する手段と、

該生成したプロセスキューを格納するプロセスキュー格納手段と、

該プロセスキューの内の実行するプロセスを前記積算手段の積算値と前記プロセス数格納手段のプロセス数とにより決定する実行プロセス決定手段を有するようにしている。

【 0 0 0 7 】

【発明の実施の形態】

最初に、本発明の実施形態についての原理的説明をする。

本発明の実施形態では、複数のプロセッサを有しそれらを結合網により結合した並列コンピュータシステムにおいて、並列度が等しい複数の並列プログラムを、時分割で多重に実行する際に、以下のスケジューリング方法によってプロセスを実行する。

各プロセッサではタイムスライス毎に、各プロセッサで $f(c, p_n)$ によって決定されるプロセスをタイムスライスの間実行する。

c は各プロセッサが有する積算カウンタの値であり、 p_n は各プロセッサで実行するプロセスの数すなわち並列プログラムの多重度であり、 $f(c, p_n)$ は c と p_n から一意に決定される。

$f(c, p_n)$ には、各プロセッサで c および p_n が等しいときには、各プロセッサで同一の並列プログラムの並列プロセスが決定されるものを用いる。

ここで、積算カウンタとは、システムの全てのプロセッサで、システムの起動時に同一の値で初期化され、全てのプロセッサで、それぞれ同期して積算されるものである。

同時刻に全てのプロセッサで積算カウンタの値 c および並列プロセスの数 p_n が等しくなり、 $f(c, p_n)$ で決定されるプロセスが同一の並列プログラムの並列プロセスとなるため、各プロセッサで実行される並列プロセスの順序が一致し、同一の並列プログラムの並列プロセスが同時に実行される。

これによって、プロセッサの相互で協調したスケジューリングに伴うオーバーヘッドを削減でき、また、プロセッサ間の明示的な通信を不必要にできる。

【0008】

また、いくつかのプロセッサを共通に使用する、並列度が異なる複数の並列プログラムを、時分割により多重に実行する際に、擬似プロセスを、該並列プログラムの並列プロセスを実行しないプロセッサにて、該並列プログラムの並列プロセスと同時に実行させ、各プロセッサで実行するプロセスの数を全てのプロセッサで均一にして、前記の同一の並列プログラムの並列プロセスを同時に実行するプロセスのスケジューリング方法を使用する。

これによって、並列度が異なる複数の並列プログラムを、時分割により多重に実行する場合にも、プロセッサの相互で協調したスケジューリングに伴うオーバーヘッドを削減でき、また、プロセッサ間の明示的な通信を不必要にできる。

【0009】

図1は、本発明によって行われるプロセスのスケジューリングの概略を示した図である。

ここではプロセッサ P_1, P_2 においてそれぞれ二つの並列プロセス A_1, A_2 と B_1, B_2 からなる並列プログラムを実行している。

プロセッサ P 1, P 2 ではそれぞれ二つのプロセス A 1, B 1 と A 2, B 2 が実行される。

時刻 T 1 ~ T 4 で各プロセッサでそれぞれ $f(c_1, p_{n1})$, $f(c_2, p_{n2})$, $f(c_3, p_{n3})$, $f(c_4, p_{n4})$ によって決定されるプロセス A 1, A 2, B 1, B 2, A 1, A 2, B 1, B 2 を実行する。

ここで、 $c_1 \sim c_4$ および $p_{n1} \sim p_{n4}$ は、それぞれ時刻 T 1 ~ T 4 の各プロセッサの積算カウンタの値とプロセス数であり、プロセッサ P 1, P 2 で等しくなる。

図 1 に示すように、同一の並列プログラムの並列プロセスが同時に実行される。

【 0 0 1 0 】

以下に、本発明の実施形態について説明する。

図 2 は、本発明の実施形態における並列コンピュータシステムの一例である。本発明の並列コンピュータシステム 1 0 0 0 では、P 1 から P N の N 個のプロセッサ群 2 0 0 0 を、結合網 3 0 0 0 によって結合している。

各プロセッサ P 1 ~ P N に、積算カウンタ 1, プロセス数カウンタ 2, プロセスキュー 3, プロセスキュー管理部 4, プロセス実行部 5 を具備している。

また、一部あるいは全てのプロセッサに、並列プログラム管理部 6 を具備している。

積算カウンタ 1 は、すべてのプロセッサで同期して積算されるもので、システムの起動時に同一の値を用いて初期化される。このようなカウンタの一例としては、公開特許公報の特開平 7 - 2 8 1 9 1 1 号に示してある、各プロセッサで有する時刻レジスタがある。

プロセス数カウンタ 2 は、各プロセッサで実行するプロセスの数を格納している。これは本システムで時分割に実行する並列プログラムの多重度と等しい。本カウンタの値は、プロセスキュー管理部 4 の処理によって増減する。

【 0 0 1 1 】

プロセスキュー 3 は、各プロセッサで実行するプロセスを格納する。説明については後述するが、プロセスキュー 3 の例を図 7, 図 8, 図 9 に示す。

一つの並列プログラムに対してシステム内で一意の識別子が決められ、同一の並列プログラムの並列プロセスには該識別子が付加される。

プロセスキュー 3 に格納されているプロセスは、該プロセスの識別子によって決定される順序にしたがって整列されている。

すなわち、同一の並列プログラムの並列プロセスは、各プロセッサのプロセスキュー 3 に同じ順序で格納されている。

本発明の実施例では、該識別子は、並列プログラム管理部 6 により付加され、その一例を後述する。

【 0 0 1 2 】

図 3 は本発明の実施形態における、並列プログラムが生成、実行、削除される流れを示した図である。

システム利用者からの並列プログラムの起動の要求 3 1 は、システム内のいずれか一つの並列プログラム管理部 6 にて処理され、並列プログラム管理部 6 から各プロセッサに対して、該並列プログラムの並列プロセスの生成要求 3 2 が出され、各プロセッサのプロセスキュー管理部 4 では、プロセスの生成要求にしたがってプロセスキュー 3 に格納し、プロセス数カウンタ 2 を増加させる (3 3)。

プロセスキュー管理部 4 によってプロセスキュー 3 に格納された、利用者が起動した並列プログラムの並列プロセスは、プロセス実行部 5 にて実行される。

プロセス実行部 5 では、各タイムスライス毎にプロセスキュー 3 に格納されているプロセスから積算カウンタ 1 とプロセス数カウンタ 2 の値によって実行するプロセスを決定しタイムスライスの間実行する (3 4)。

並列プログラムのすべての並列プロセスの処理が終了するまで、プロセス実行部 5 によって時分割に繰り返し実行される。

並列プログラムの並列プロセスのすべての処理が終了すると (3 5)、並列プログラム管理部 6 によって、各プロセッサに対して、該並列プログラムの並列プロセスの削除要求 3 6 が出され、各プロセッサのプロセスキュー管理部 4 では、削除要求にしたがってプロセスキュー 3 から削除し、プロセス数カウンタ 2 を減少させる (3 7)。

【 0 0 1 3 】

本発明の第一の実施形態を図 2, 図 4, 図 5, 図 7, 図 8, 図 10, 図 11, 図 12 を用いて説明する。

並列プログラムの並列プロセスは、並列コンピュータシステム 1000 の各プロセッサのプロセス実行部 5 で実行する。

プロセスの実行がタイムスライスを使い切った場合、実行中のプロセスを中断し、プロセスキュー 3 から、積算カウンタ 1 およびプロセス数カウンタ 2 により、次に実行するプロセスを決定し、そのプロセスを実行する。

プロセス実行部 5 はタイムスライス毎に実行するプロセスを決定し実行するので、並列プログラムの各々の並列プロセスは、各プロセッサのプロセス実行部 5 によって実行される順序が決定される。

【0014】

図 10 は、プロセス実行部 5 の処理を示したフローチャートであり、以下のようになっている。

なお、本処理は、システムが定期的に発生するタイマ割り込みの処理にて起動される。

一例では、タイマー割り込みの間隔は 10msec であり、タイムスライスは 100msec である。

ステップ 101 では、該プロセッサにおいて現在実行中のプロセスがタイムスライスを使い切ったかどうかを判定し、使い切っていない場合は処理を終了し、使い切った場合は処理をステップ 102 に移す。

ステップ 102 では、積算カウンタ 1 とプロセス数カウンタ 2 によって、プロセスキュー 3 から実行するプロセスを決定し、処理をステップ 103 へ移す。

なお、実行するプロセスを決定する方法の一例として、プロセスキュー 3 の先頭から、「(積算カウンタ 1 の値/タイムスライス) mod プロセス数カウンタ 2 の値」番目のプロセスを、実行するプロセスとする。

ここで「 P/Q 」は「 P を Q で割った商」の意味であり、「 $R \bmod S$ 」は R を S で割った余り」の意味である。また、ステップ 102 の処理は、プロセスキュー管理部 4 の処理と排他される。

ステップ 103 では、次に実行するプロセスへの切換え処理が必要かどうかを

判定し、必要な場合は処理をステップ 1 0 4 へ移し、必要ない場合は処理を終了する。

なお、プロセスの切換えが必要かどうかの判定は、ステップ 1 0 2 により決定されたプロセスと、前回実行していたプロセスが、同一であるかにより行われる。

ステップ 1 0 4 では、ステップ 1 0 2 により決定されたプロセスへ、コンテキストを切換え、該プロセスの実行を開始する。その後、処理を終了する。

以上のようにして、並列度が等しい複数の並列プログラムの各々の並列プロセスは、タイムスライス毎に各プロセッサのプロセス実行部 5 にて実行するプロセスを決定され実行される。

すなわち、各プロセッサのプロセス実行部 5 にて、同一の並列プログラムの全ての並列プロセスを同時に実行するように、プロセスの実行順序を決定している。

【 0 0 1 5 】

図 1 1 は、プロセスキュー管理部 4 の処理を示したフローチャートであり、以下のようにになっている。

なお、本処理は、システム内のいずれかの並列プログラム管理部 6 のプロセスの生成あるいは削除の処理により起動される。同時に複数の並列プログラム管理部 6 の処理によってプロセス生成あるいは削除が行われる場合は、プロセスキュー管理部 4 は、順次その処理を行い、その他の処理と排他される。

また、プロセス実行部 5 が実行するプロセスを決定する場合も、プロセスキュー管理部 4 はプロセス実行部 5 と排他して動作する。

ステップ 1 1 1 では、プロセスの生成か削除かを判定する。プロセス生成の場合は処理をステップ 1 1 2 へ移し、プロセス削除の場合は処理をステップ 1 1 4 へ移す。

ステップ 1 1 2 では、プロセスをプロセスキュー 3 に、プロセスに付加されている識別子により整列されるように格納し、処理をステップ 1 1 3 へ移す。

ステップ 1 1 3 では、プロセス数カウンタ 2 を 1 増やし、処理を終了する。

ステップ 1 1 4 では、プロセスをプロセスキュー 3 から削除し、処理をステッ

プ 1 1 5 へ移す。

ステップ 1 1 5 では、プロセス数カウンタ 2 を 1 減らし、処理を終了する。

【 0 0 1 6 】

図 1 2 は、並列プログラム管理部 6 の処理を示したフローチャートであり、以下のようになっている。

なお、並列プログラム管理部 6 は、システムの利用者の並列プログラム生成の要求あるいは一つの並列プログラムのすべての並列プロセスの処理が終了した場合に起動される。

また、同一の並列プログラム管理部 6 に対して、複数の並列プログラムの生成の要求がある場合は、該プロセッサにおいて、同時に、並列プログラム管理部での複数の処理が行われる。

ステップ 1 2 1 では、並列プログラムの生成か削除かを判定し、生成の場合はステップ 1 2 2 へ処理を移し、削除の場合はステップ 1 2 4 へ処理を移す。

ステップ 1 2 2 では、該並列プログラムに対しシステム内で一意となる識別子を決定し、該並列プログラムの並列プロセスに対して該識別子を付加する。その後、処理をステップ 1 2 3 へ移す。

一例として、識別子は以下を用いる。

各プロセッサ P_i ($i = 1 \sim N$) に具備された並列プログラム管理部 6 では、並列プログラムに 0 ~ 9 9 9 9 の番号を付加する。

この番号は、同一の並列プログラム管理部 6 で生成され実行中の並列プログラムでは重複しないものを用いる。

各並列プログラム管理部 6 では、前記番号と「 $10000 \times i$ 」を加算した値を該並列プログラムの並列プロセスの識別子として付加する。

ステップ 1 2 3 では、並列プログラムの生成の要求によって、該並列プログラムの並列プロセスを実行するプロセッサに対して、当該並列プログラムの並列プロセスを生成し、処理を終了する。

ステップ 1 2 4 では、該並列プログラムの並列プロセスを実行した、すべてのプロセッサに対して、当該並列プログラムの並列プロセスの削除を行い、処理を終了する。

【 0 0 1 7 】

図 4 , 図 5 は、本発明の第一の実施形態における、複数の並列プログラムの並列プロセスを時分割に実行したときの、各プロセッサで実行されるプロセスの時間による遷移の一例を示した図である。

この例では、各並列プログラムはプロセッサ P 1 , P 2 の二つのプロセッサを使用する。また、タイムスライスを 100msec としており、積算カウンタとしては時刻レジスタを用いている。

図 4 の例では、並列プログラムは 7 1 , 7 2 の二つであり、それぞれ二つのプロセス A 1 , A 2 と B 1 , B 2 からなる。

各プロセッサのプロセスキュー 3 は図 7 のようになっており、プロセッサ P 1 , P 2 のプロセスキュー 3 には、それぞれプロセス A 1 , B 1 、 A 2 , B 2 が格納されている。

また、プロセス A 1 , A 2 には識別子 x , B 1 , B 2 には識別子 y が付加されており、各プロセッサのプロセスキュー 3 に格納されている。

ここで、プロセスキュー 3 はプロセスに付加されている識別子 x と y によって整列されるため、同一の並列プログラムの並列プロセスの順序が、プロセッサ P 1 , P 2 のそれぞれのプロセスキュー 3 において等しくなる。

この例では、識別子 x , y の順序となるように整列される。

時刻 4 1 では、「 $(0/100) \bmod 2 = 0$ 」であるので、プロセスキューの先頭から 0 番めの並列プログラム 7 1 のプロセス A 1 , A 2 が、それぞれのプロセッサで実行される。

同様にして各プロセッサで実行するプロセスを決定し実行する。時刻 4 2 , 4 3 , 4 4 にて、それぞれ並列プログラム 7 2 , 7 1 , 7 2 のプロセス B 1 , B 2 、 A 1 , A 2 、 B 1 , B 2 が実行される。

【 0 0 1 8 】

図 5 は、本発明のプロセスのスケジューリング方法により、並列度が等しい二つの並列プログラムが時分割に多重に実行されていた場合に、新たに並列度が等しい一つの並列プログラムが生成された際の、各プロセッサで実行されるプロセスの時間による遷移の一例を示した図である。

一つの並列プログラムの並列プロセスは、必ずしも、該並列プログラムの並列プロセスを実行する全てのプロセッサで同時に生成、削除が行われない。

図 5 の例では、時刻 5 1 に、プロセッサ P 1 ではプロセス数は 3 であるが、P 2 では 2 である。

一時的に該並列プログラムの並列プロセスを実行するプロセッサでのプロセス数が同一でないために、プロセスの実行順序が変わり、同一の並列プログラムの並列プロセスが同時に実行されないことがあるが、以下の例に示すように、プロセッサ間の通信を行わなくとも、最終的に同一の並列プログラムの並列プロセスが同時に実行される。

実行中の並列プログラムが 7 1, 7 2, 7 3 の三つであり、それぞれ二つのプロセス A 1, A 2, B 1, B 2, C 1, C 2 からなるときの、各プロセッサのプロセスキューの例を図 8 に示す。

プロセッサ P 1, P 2 のプロセスキュー 3 には、それぞれプロセス A 1, B 1, C 1, A 2, B 2, C 2 が格納されている。

また、プロセス A 1, A 2 には識別子 x、B 1, B 2 には識別子 y、C 1, C 2 には識別子 z が付加されており、各プロセッサのプロセスキュー 3 に格納されている。

ここで、プロセスキュー 3 はプロセスに付加されている識別子 x, y, z によって整列されるため、同一の並列プログラムの並列プロセスの順序が、プロセッサ P 1, P 2 のそれぞれのプロセスキュー 3 において等しくなる。

この例では、識別子 x, y, z の順序となるように整列される。

【 0 0 1 9 】

図 5 の例では、時刻 5 1 ではプロセッサ P 1 のプロセスキューには、図 8 のように、A 1, B 1, C 1 の三つのプロセスを格納しているが、プロセッサ P 2 のプロセスキューは、図 7 のように、A 2, B 2 の二つのプロセスを格納している。

時刻 5 1 では、プロセッサ P 1 では、「 $(500/100) \bmod 3 = 2$ 」であるので、並列プログラム 5 3 のプロセス C 1 が実行される。

プロセッサ P 2 では、「 $(500/100) \bmod 2 = 1$ 」であるので、並列プログラム 7 2 のプロセス B 2 が実行される。

時刻 5 2 では、プロセッサ P 1 , P 2 でプロセス数が 3 となり、各プロセッサのプロセスキューは図 8 のようになる。

時刻 5 2 では、「 $(600/100) \bmod 3 = 0$ 」であるので、並列プログラム 7 1 のプロセス A 1 , A 2 がそれぞれ実行される。

同様に、時刻 5 3 , 5 4 にて、それぞれ並列プログラム 7 2 , 7 3 のプロセス B 1 , B 2 , C 1 , C 2 が実行される。

このように、並列プログラムの並列プロセスの生成や削除によって、一時的に各プロセッサでのプロセス数が一致しない場合でも、プロセッサ間の通信を行わずに、最終的には、同一の並列プログラムの並列プロセスを同時に実行するプロセスのスケジューリングが行える。

【 0 0 2 0 】

本発明の第二の実施形態では、並列プログラムの並列度が異なる場合には、並列プログラム管理部 6 の生成処理において、擬似プロセスを生成し、該並列プログラムの並列プロセスを実行しないプロセッサにおいて、該並列プログラムの並列プロセスと同時に実行する。

本発明の第二の実施形態を図 2 , 図 6 , 図 9 , 図 1 3 を用いて説明する。

なお、プロセスキュー管理部 4 とプロセス実行部 5 の処理は第一の実施形態と同様である。

すなわち、第一の実施形態と同様に、並列度が異なる並列プログラムの各々の並列プロセスは、各プロセッサのプロセス実行部 5 にて、同一の並列プログラムのすべての並列プロセスを同時に実行するように、プロセスの実行順序を決定される。

図 1 3 は本発明の第二の実施形態における、並列プログラム管理部 6 の処理を示したフローチャートであり、以下のようになっている。

ステップ 1 3 1 では、並列プログラムの生成か削除かを判定し、生成の場合はステップ 1 3 2 へ処理を移し、削除の場合はステップ 1 3 4 へ処理を移す。

ステップ 1 3 2 では、該並列プログラムの並列プロセスに対して、システム内で一意となる識別子を付加し、処理をステップ 1 3 3 に移す。

ステップ 1 3 3 では、生成する並列プログラムの並列度が、時分割で多重に実

行する並列プログラムで使用するプロセッサの数と比べて小さい場合には、擬似プロセスを必要な数だけ生成し、また、並列プログラムの生成要求にしたがって、各プロセッサに対して生成し、処理を終了する。

擬似プロセスは、何も行わず、単にスピループするようなものでよい。

また、並列プログラムの並列度は、該並列プログラムのコンパイル時あるいは起動時に指定されるものであり、これによって擬似プロセスの数を求めることができる。

また、擬似プロセスにも、同時に実行する並列プログラムの並列プロセスと同一の識別子が付加される。

ステップ 1 3 4 では、該並列プログラムの並列プロセスと擬似プロセスを実行した、すべてのプロセッサに対して、該並列プログラムの並列プロセスおよび該並列プログラムの並列プロセスと同時に実行した擬似プロセスの削除を行い、処理を終了する。

【 0 0 2 1 】

図 6 は、本発明の第二の実施形態における、並列度の異なる複数の並列プログラムを時分割により多重に実行する際に、擬似プロセスを実行させた場合の、各プロセッサで実行するプロセスの時間による遷移の一例を示した図である。

この例では、並列プログラムは 7 1 , 7 4 の二つであり、並列プログラム 7 1 は二つのプロセス A 1 , A 2 からなり、並列プログラム 7 4 は一つのプロセス D 1 からなり、並列プログラム 7 1 , 7 4 はプロセッサ P 1 を共通に使用する。

各プロセッサのプロセスキューは、図 9 のようになっており、プロセッサ P 1 のプロセスキューには、プロセス A 1 , D 1 が格納されており、プロセッサ P 2 のプロセスキューには A 2 と擬似プロセスが格納されている。

また、プロセス A 1 , A 2 には識別子 x , D 1 と擬似プロセスには識別子 w が付加されており、各プロセッサのプロセスキュー 3 に格納されている。プロセスキュー 3 はプロセスに付加されている識別子 x と w によって整列されるため、同一の並列プログラムの並列プロセスの順序が、プロセッサ P 1 , P 2 のそれぞれのプロセスキューにおいて等しくなる。

この例では、識別子 x , w の順序となるように整列される。

時刻 6 1, 6 2, 6 3 にて、それぞれ「 $(0/100) \bmod 2 = 0$ 」, 「 $(100/100) \bmod 2 = 1$ 」, 「 $(200/100) \bmod 2 = 0$ 」であるので、並列プログラム 7 1, 7 4, 7 1 のプロセス A 1, A 2、D 1、A 1, A 2 が実行される。

また、プロセッサ P 2 では、時刻 6 2 で擬似プロセスが実行される。

このように、並列度が異なる複数の並列プログラムを時分割により実行する場合には、一つの並列プログラムの並列プロセスを実行しないプロセッサにおいて、擬似プロセスを該並列プログラムの並列プロセスと同時に実行することによって、同一の並列プログラムの並列プロセスを同時に実行するプロセスのスケジューリングが行える。

【 0 0 2 2 】

【発明の効果】

以上説明したように、本発明によれば、プロセッサの相互で協調したスケジューリングに伴うプロセッサ間のオーバーヘッドを削減した、プロセッサ間の明示的な通信を必要としない、プロセスのスケジューリングが行える。

【図面の簡単な説明】

【図 1】

本発明のスケジューリング方法により、各プロセッサで実行されるプロセスの時間による遷移の概略を示した図である。

【図 2】

本発明の実施形態における並列コンピュータシステムの一例を示した図である。

【図 3】

本発明の実施形態における並列プログラムの動作の流れを示した図である。

【図 4】 本発明の第一の実施形態における、スケジューリング方法により、各プロセッサで実行されるプロセスの、時間による遷移の一例を示した図である。

【図 5】

本発明の第一の実施形態における、スケジューリング方法により、並列プログラムの並列プロセスが生成された場合の、各プロセッサで実行されるプロセスの

、時間による遷移の一例を示した図である。

【図 6】

本発明の第二の実施形態における、スケジューリング方法により、各プロセッサで実行されるプロセスの、時間による遷移の一例を示した図である。

【図 7】

本発明の第一の実施形態における、プロセス数が 2 のときの、プロセスキューの一例を示した図である。

【図 8】

本発明の第一の実施形態における、プロセス数が 3 のときの、プロセスキューの一例を示した図である。

【図 9】

本発明の第二の実施形態における、プロセス数が 2 のときの、プロセスキューの一例を示した図である。

【図 1 0】

本発明の実施形態における、プロセス実行部の処理を示したフローチャートである。

【図 1 1】

本発明の実施形態における、プロセスキュー管理部の処理を示したフローチャートである。

【図 1 2】

本発明の第一の実施形態における、並列プログラム管理部の処理を示したフローチャートである。

【図 1 3】

本発明の第二の実施形態における、並列プログラム管理部の処理を示したフローチャートである。

【符号の説明】

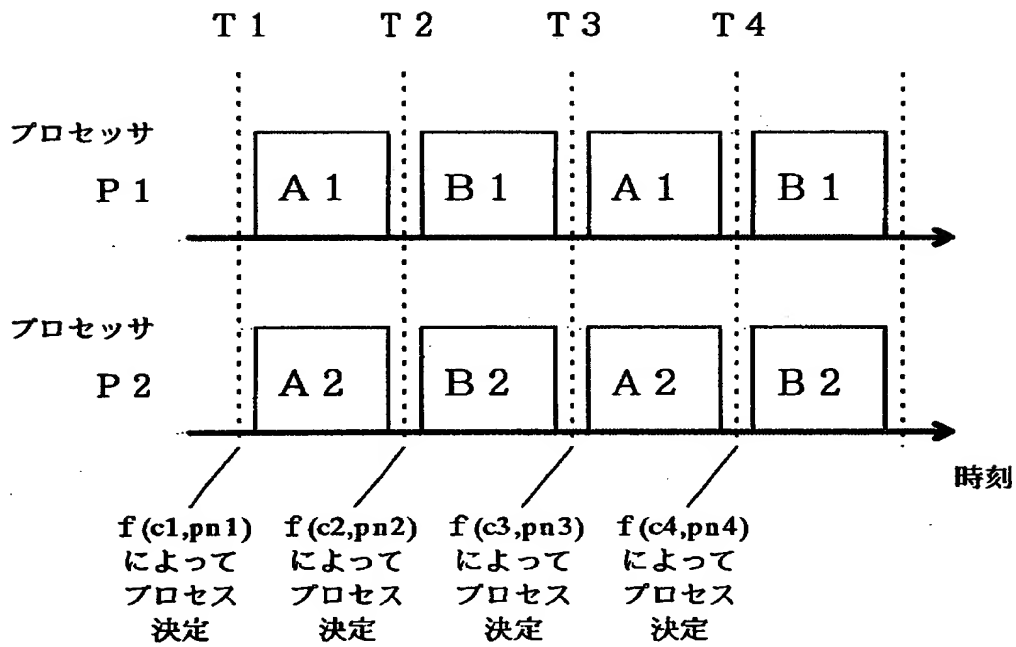
- 1 積算カウンタ
- 2 プロセス数カウンタ
- 3 プロセスキュー

- 4 プロセスキュー管理部
- 5 プロセス実行部
- 6 並列プログラム管理部
- 1 0 0 0 並列コンピュータシステム
- 2 0 0 0 プロセッサ
- 3 0 0 0 プロセッサ間結合網

【書類名】 図面

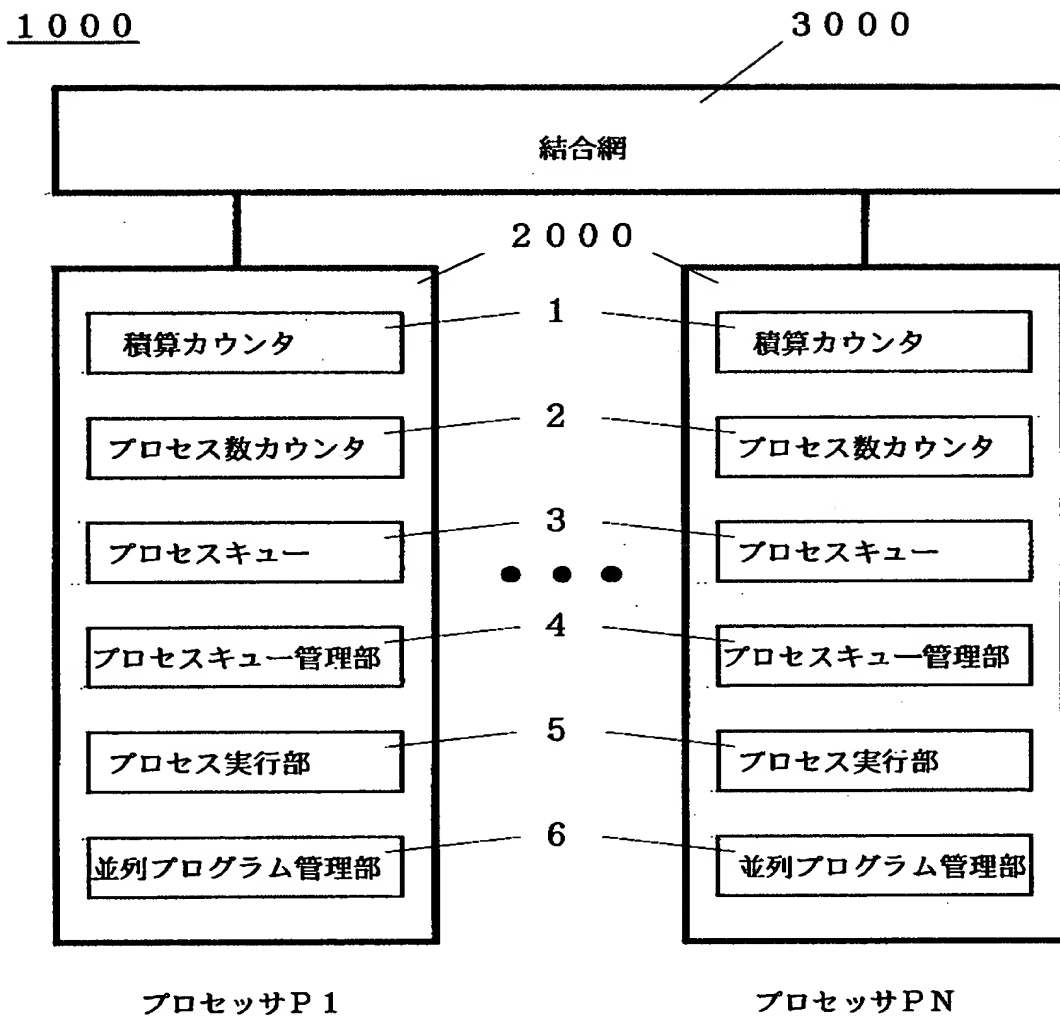
【図1】

【図1】



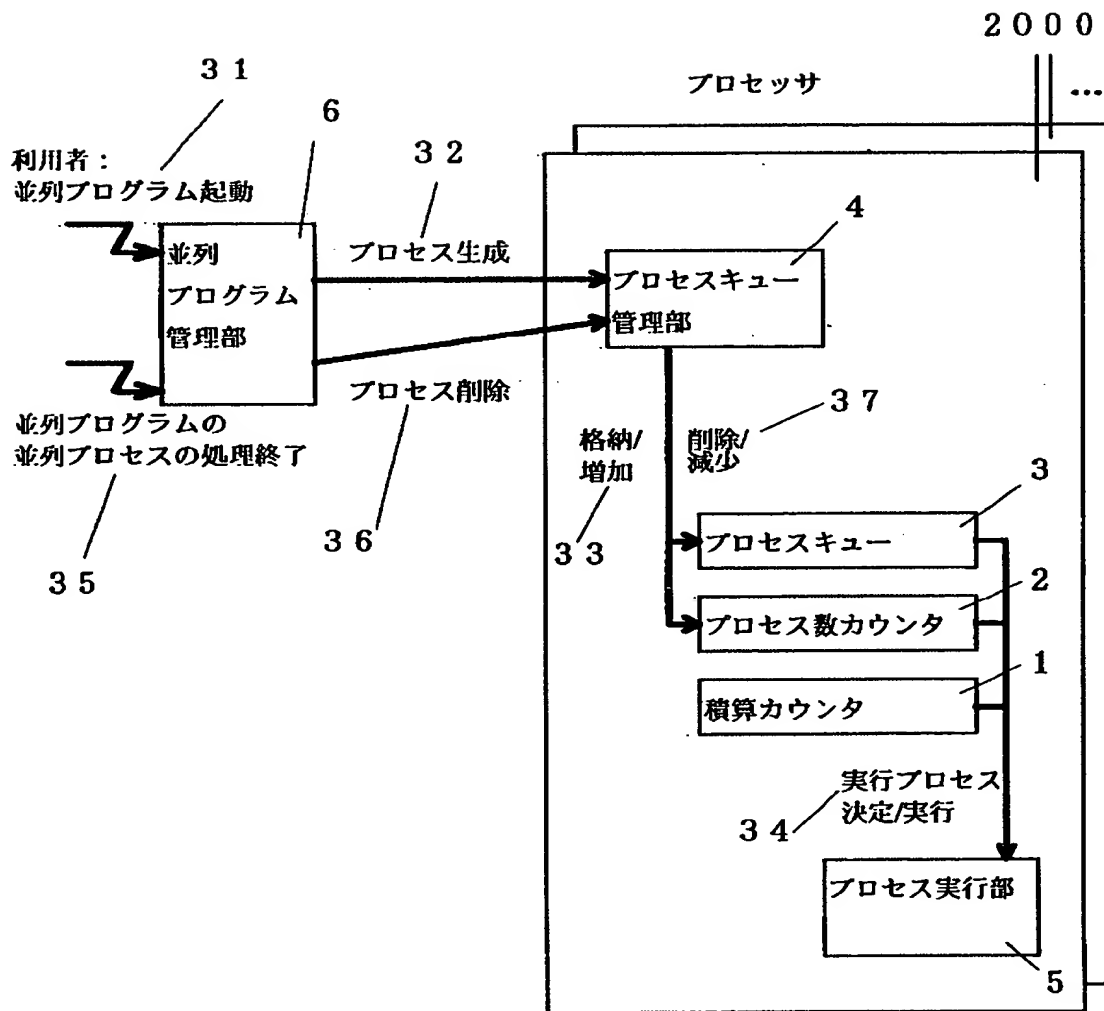
【図 2】

【図 2】

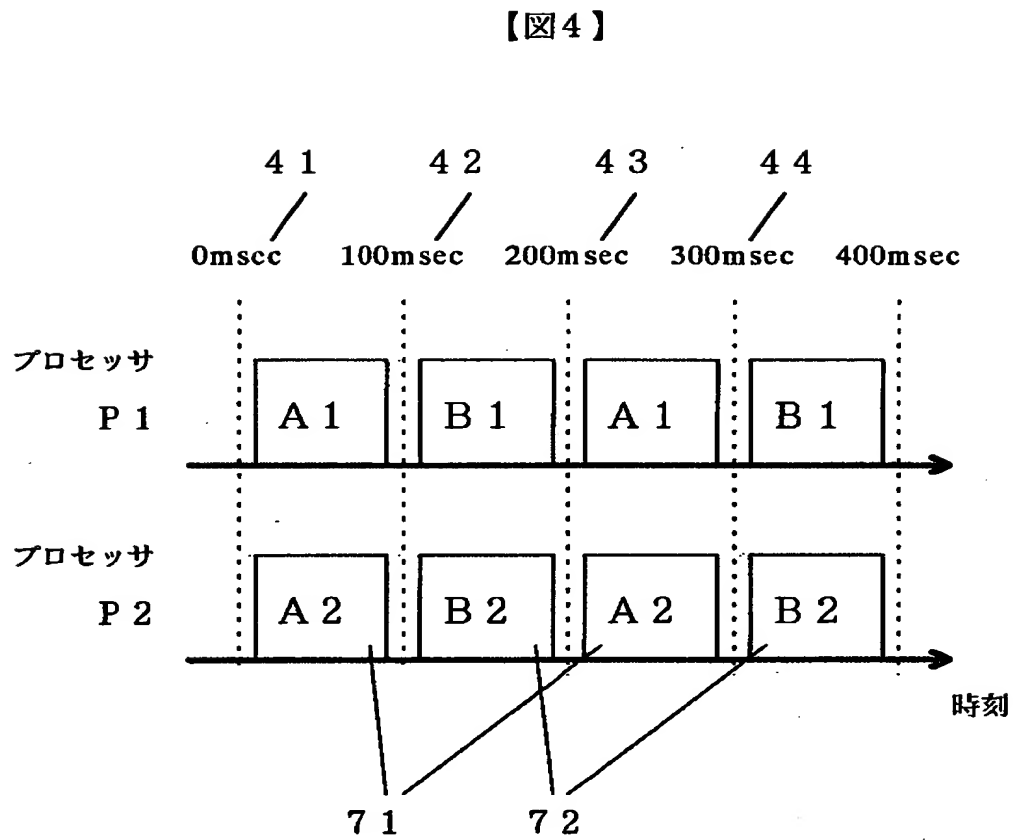


【図3】

【図3】

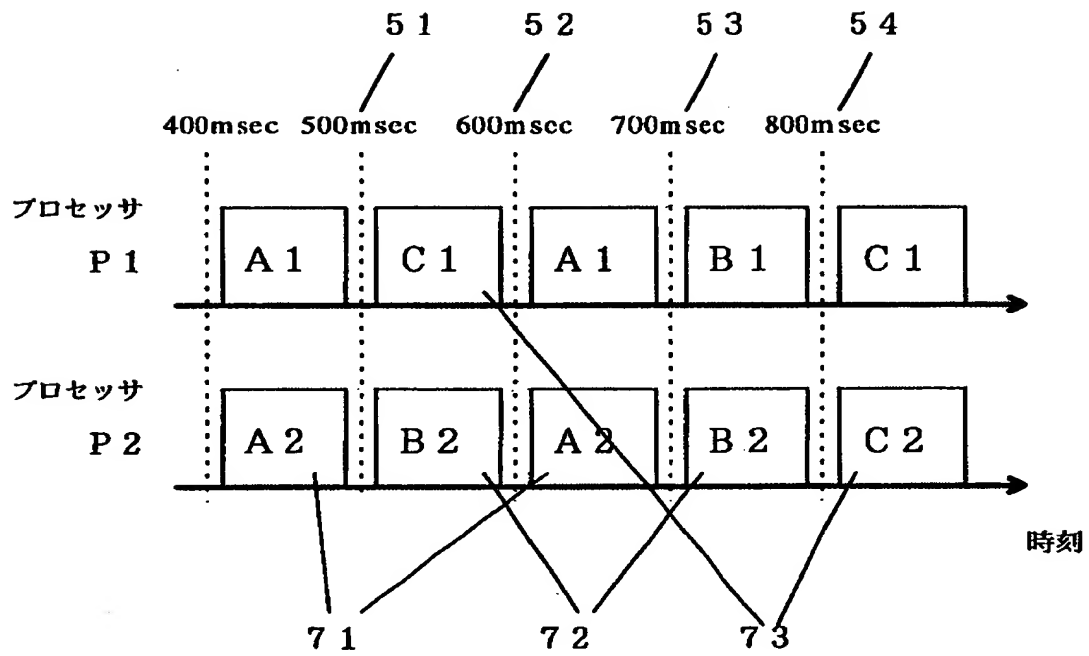


【図4】



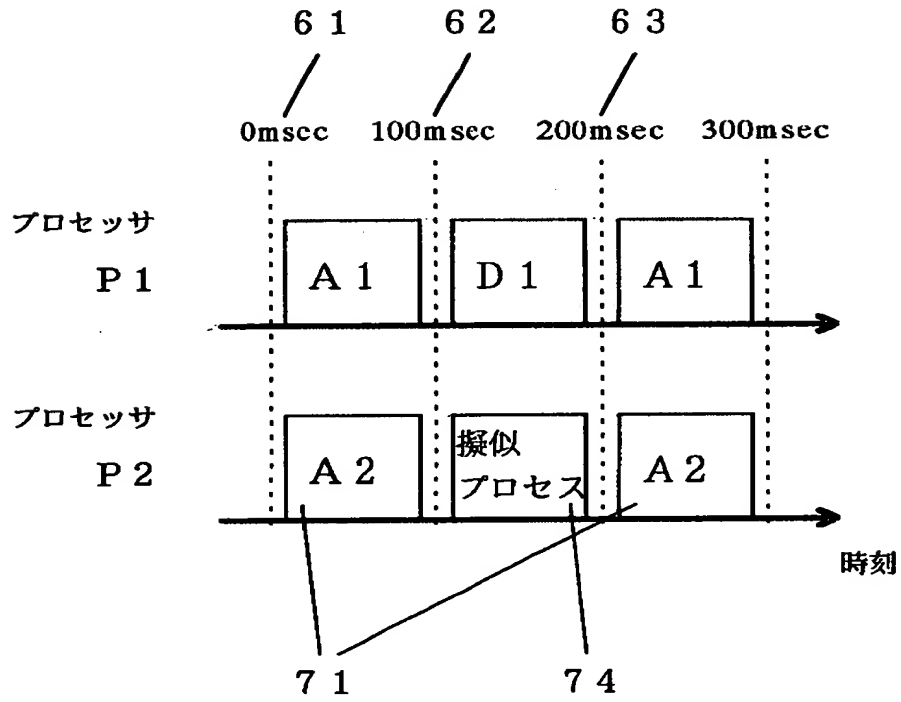
【図 5】

【図 5】



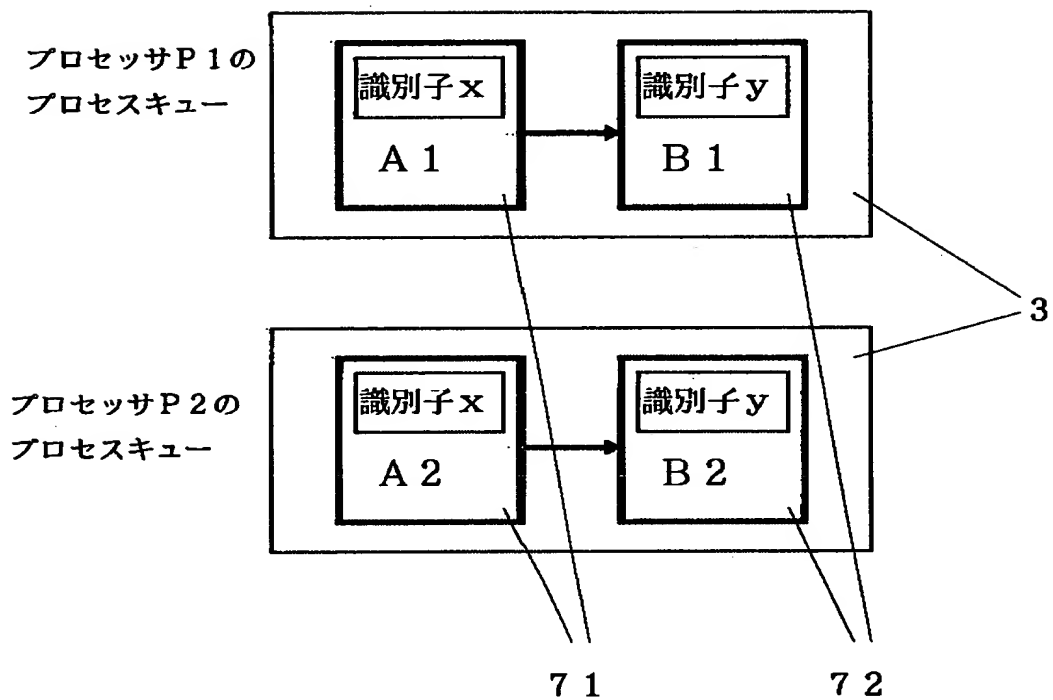
【図 6】

【図 6】



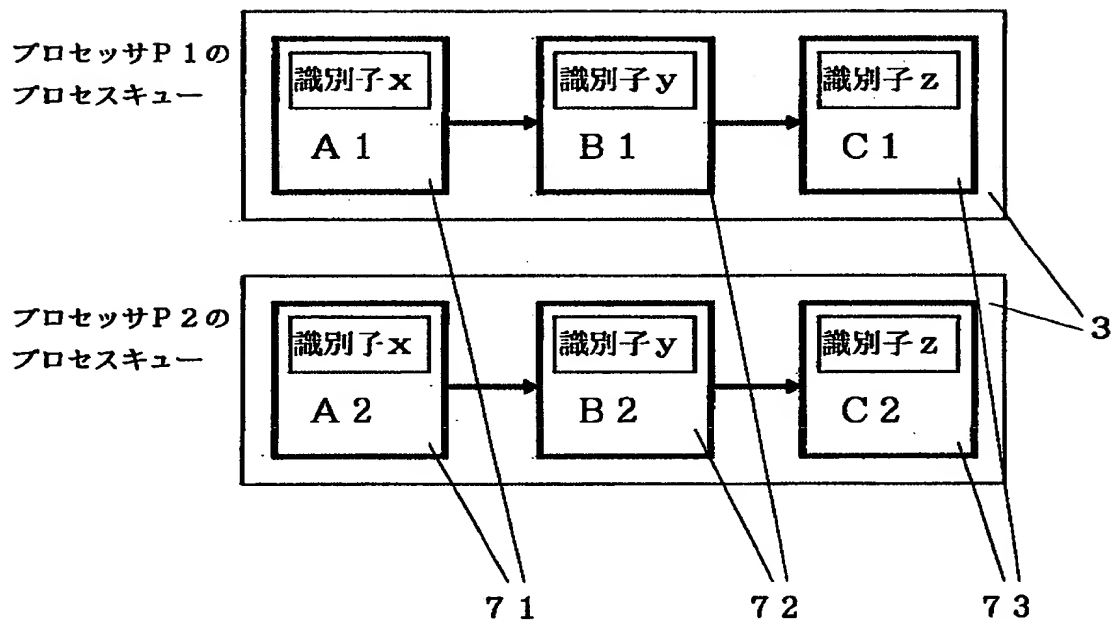
【図 7】

【図 7】



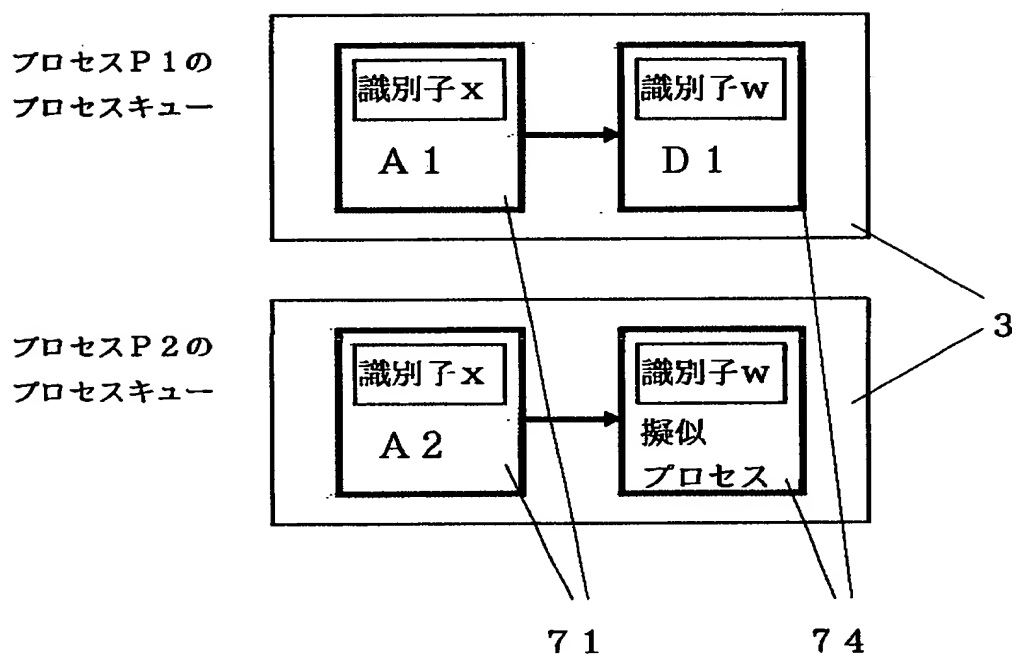
【図 8】

【図 8】

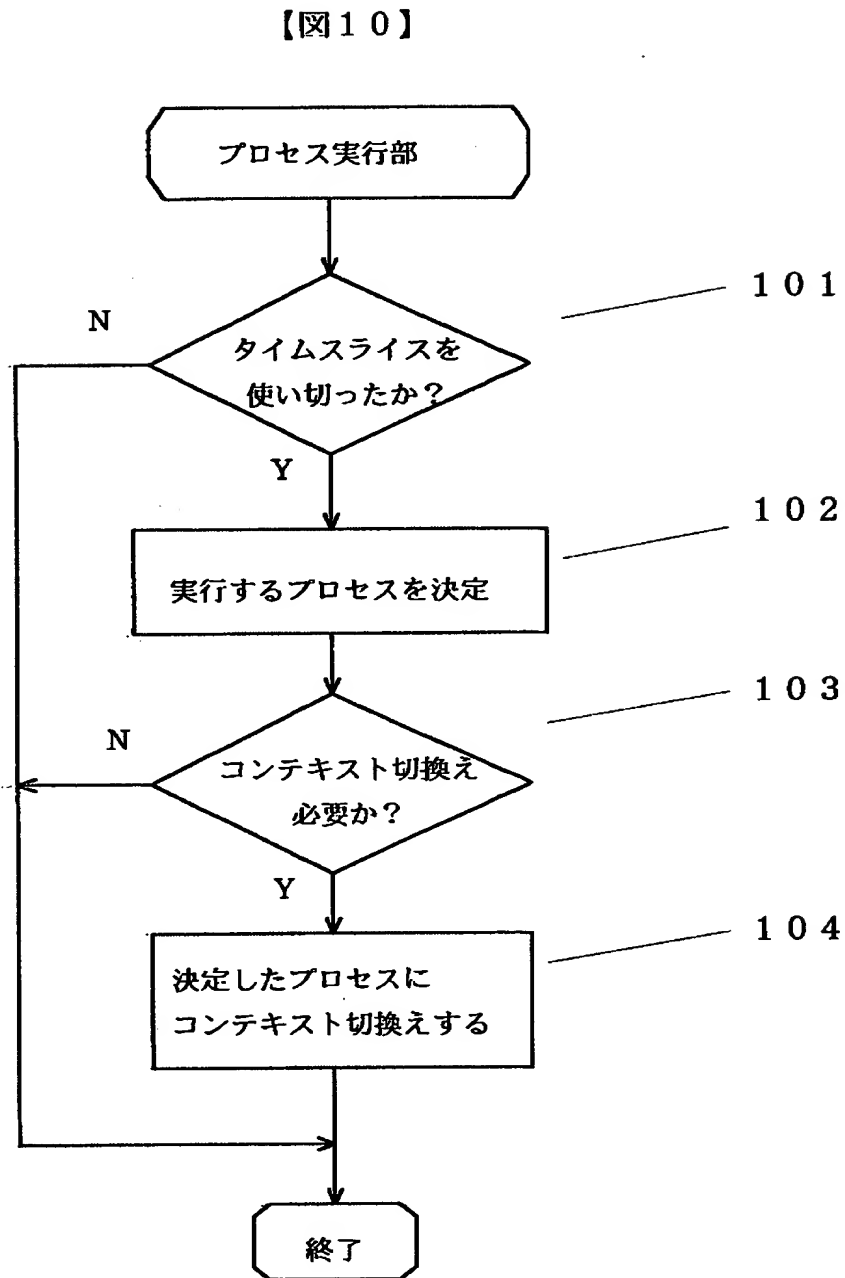


【図 9】

【図 9】

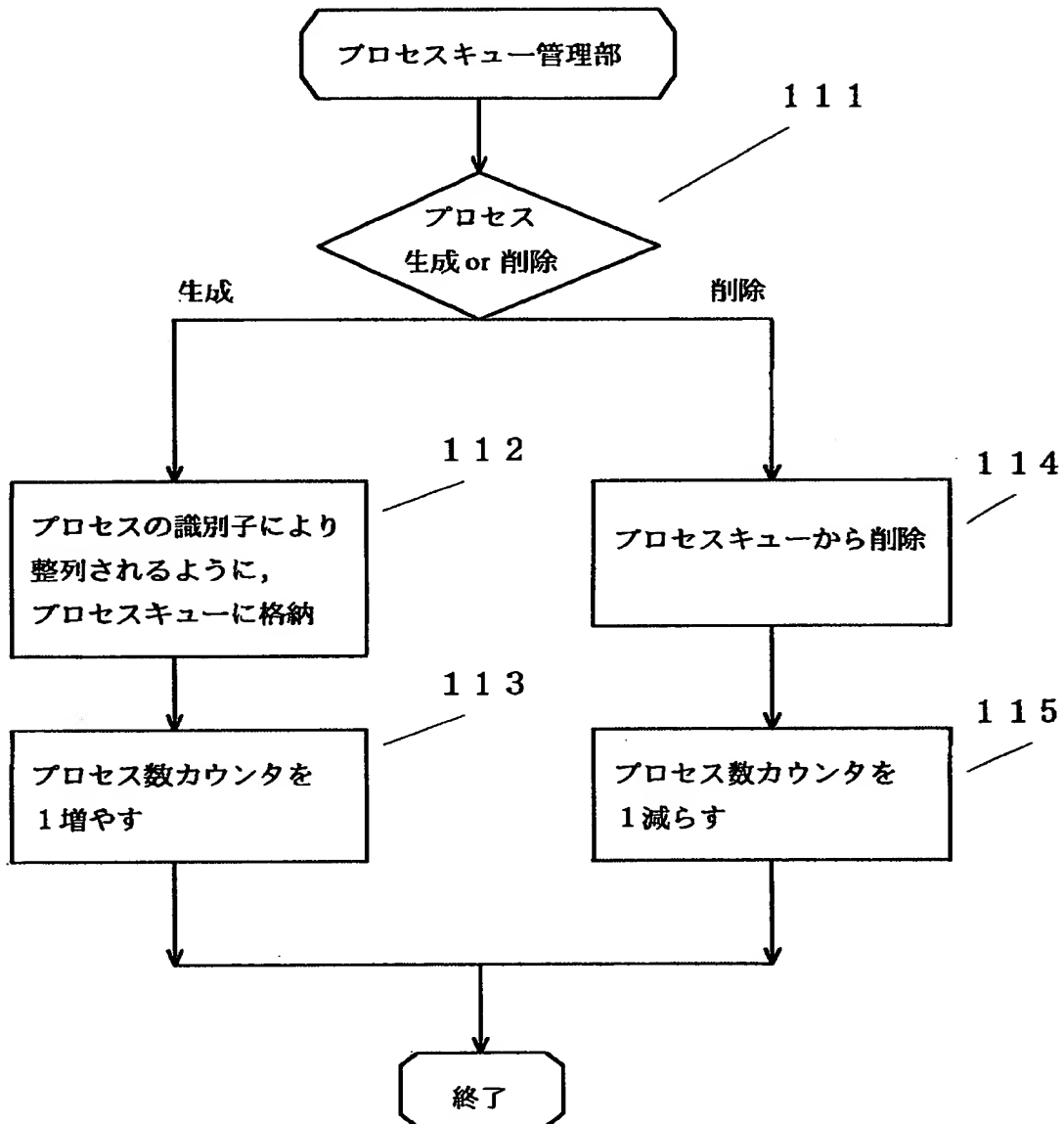


【図 1 0】

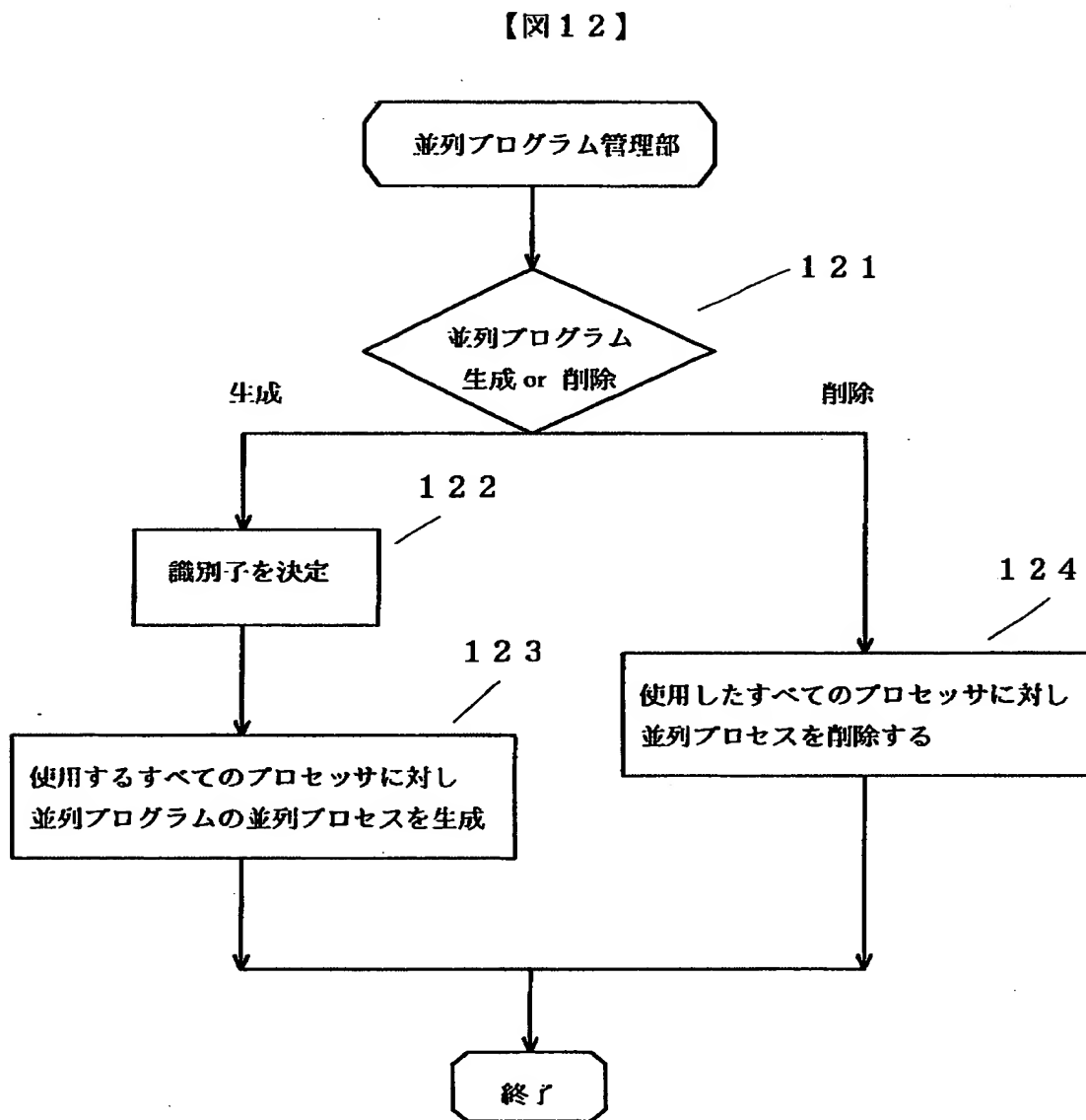


【図 11】

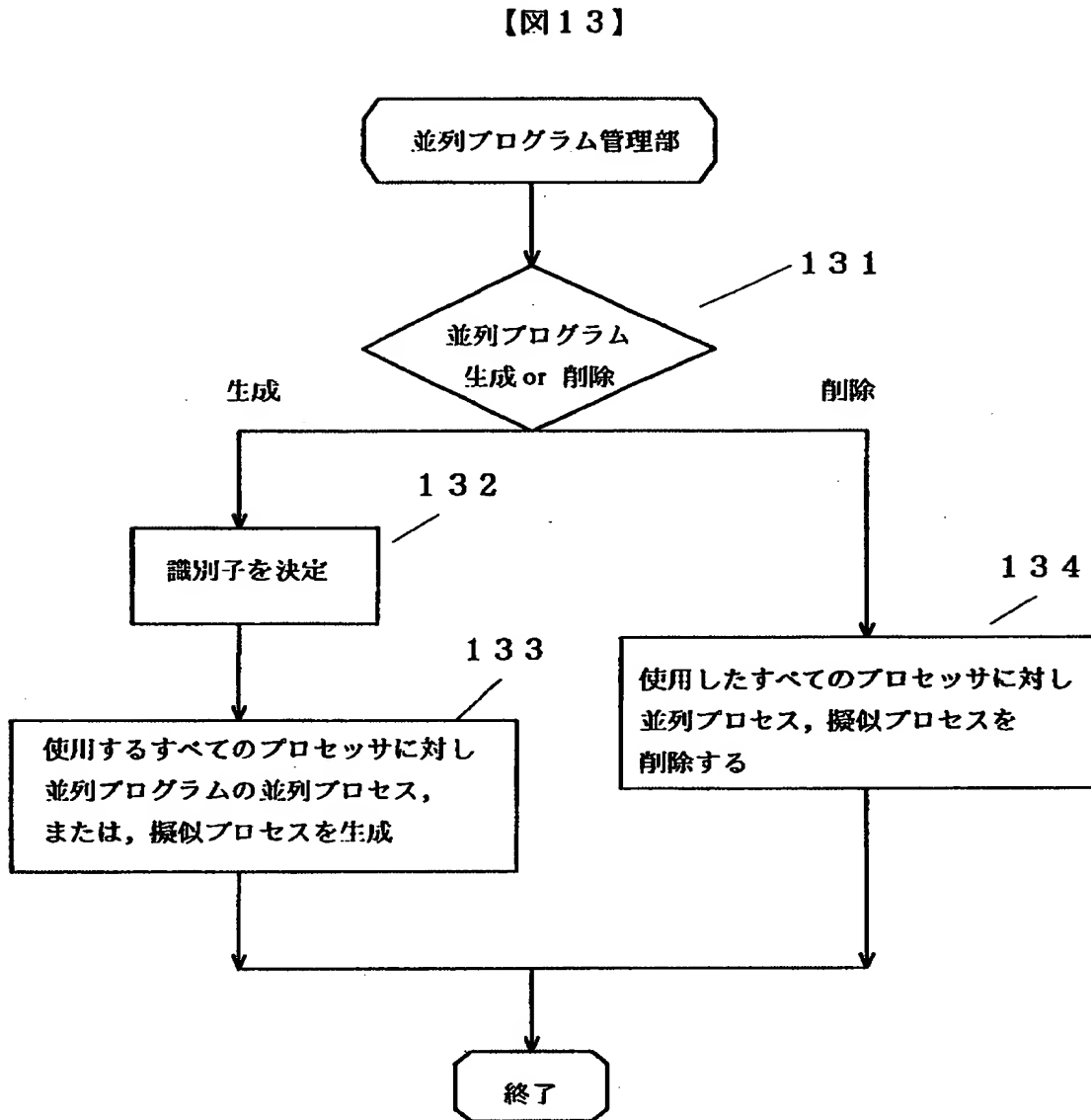
【図 11】



【図 12】



【図13】



【書類名】 要約書

【要約】

【課題】 プロセッサの相互で協調したスケジューリングに伴うオーバーヘッドを削減した、プロセッサ間の明示的な通信を必要としない、プロセスのスケジューリング方法を提供することにある。

【解決手段】 並列プログラムの起動又は終了に応じて並列プログラム管理部6はプロセスの生成又は削除をプロセスキュー管理部4に指示し、管理部4は該指示に従いプロセスキュー3へプロセスを格納又はプロセスキュー3からプロセスを削除し、プロセスカウンタ2は起動された並列プログラムの数に対応するプロセスの数を記憶し、積算カウンタ1は全てのプロセッサで同期して積算され、プロセスキュー3の内の実行されるプロセスは、プロセスキュー3の先頭から、「(積算カウンタ1の値/タイムスライス) mod プロセス数カウンタ2の値」番目のプロセスとする。上記タイムスライスとはプロセスが実行されるタイムスライスである。

【選択図】 図3

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台4丁目6番地
氏 名 株式会社日立製作所